力测量平台

信息展示与交互系统

姓　　名：	金大为
学　　号：	2020011054
完成日期：	2022 年 7 月

# 初始用户名：admin，初始密码：111111（为了防止打不开写在最前面）

【用户文档】

(1) 应用程序功能的详细说明：

1. 从 excel 库读取力与时间的数据。
2. 用户登录和新用户注册（注册时对用户名密码进行合法性检查）
3. 读取用户数据（姓名，身高，体重）
4. 根据身高体重自动计算 BMI
5. 编辑更改用户的信息（姓名，身高，体重），更改密码（对编辑数据进行合法性检查）
6. 在界面表格中显示用户数据
7. 将获取的力测量数据通过动态数据流方式输入并通过动态曲线图展示出来，力数据实时获取和显示
8. 启动和暂停模拟数据流输入
9. 通过力测量数据计算相关参数并显示在表格中
10. 将获取的反跳数据通过曲线图展示
11. 自动在图中标注相关的参数

12. 保存参数到 excel 表

13. 保存绘制的数据图像

(2) 应用程序运行环境的要求：

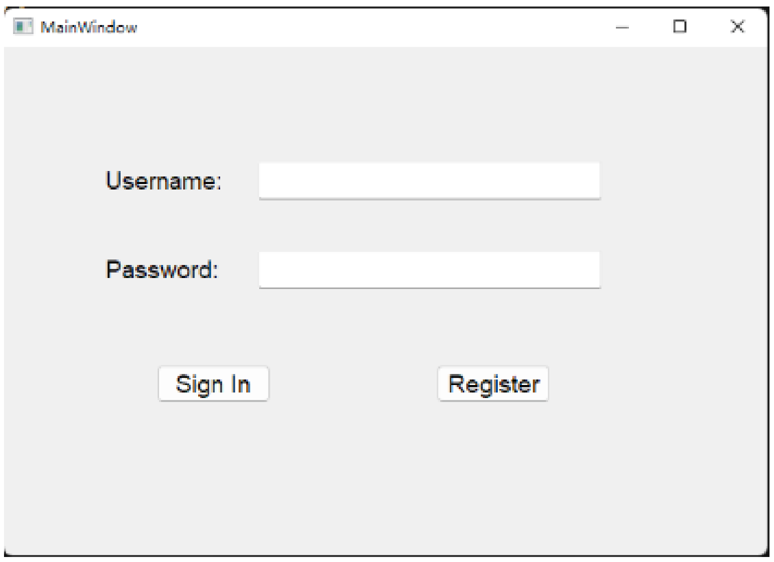
Python3.7，要求安装 matplotlib，pyqt5，numpy，pandas 库

(2) 应用程序的启动方法：

有 python 编译器并安装了 matplotlib，pyqt5，numpy，pandas 库的可以直接双击 final.py 即可启动，库不全的需要 pip install 相关的库

(4) 程序的界面、交互方式和操作方法

1. 登录界面：

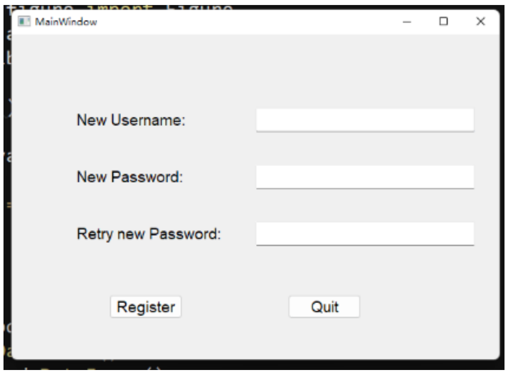


在 Username 中输入用户名，在 Password 中输入密码

默认账户为：admin 默认密码为：111111

点击 Sign in 后，用户名密码正确就会跳转到主页面，不正确会提示用户名密码不正确，可点击 Register 注册新的账号。

2. 注册界面：

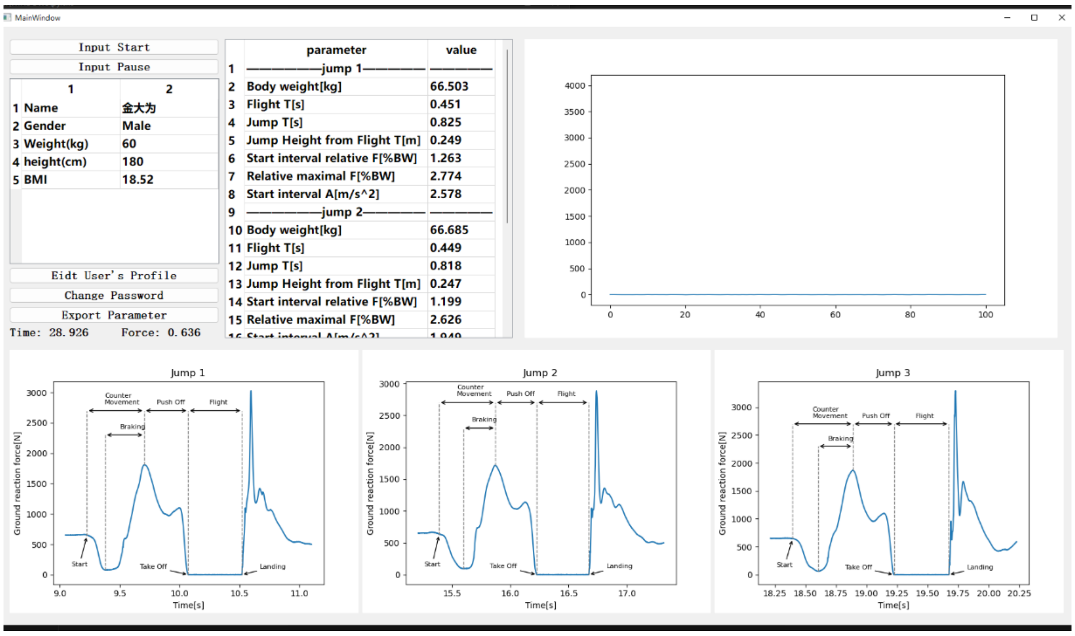


输入用户名，密码和确认密码，点击 Register，

如果输入的不合法会有相关的提示指出其为哪种不合法性，如合法即会显示注册成功，可回到登陆界面登陆。

点击 cancel 可以取消注册

3. 主界面



	1	2
1	Name	金大为
2	Gender	Male
3	Weight(kg)	60
4	height(cm)	180
5	BMI	18.52

Time: 28.926   Force: 0.636

	parameter	value
1	————————jump 1————	————
2	Body weight[kg]	66.503
3	Flight T[s]	0.451
4	Jump T[s]	0.825
5	Jump Height from Flight T[m]	0.249
6	Start interval relative F[%BW]	1.263
7	Relative maximal F[%BW]	2.774
8	Start interval A[m/s^2]	2.578
9	————————jump 2————	————
10	Body weight[kg]	66.685
11	Flight T[s]	0.449
12	Jump T[s]	0.818
13	Jump Height from Flight T[m]	0.247
14	Start interval relative F[%BW]	1.199
15	Relative maximal F[%BW]	2.626

i. 左上角为操作面版和用户信息显示区域

   a. 点击 Input Start 可以开始模拟数据流输入

   b. 点击 Input Stop 可以暂停模拟数据流输入

   c. 中间的表格显示用户的数据

   d. 点击 Edit Profile 会跳出用户信息更改界面

   e. 点击 Change Password 会跳转到更改密码界面

   f. 点击 Export parameter 会在同文件夹中新建 Excel 表格并保存计算的参数

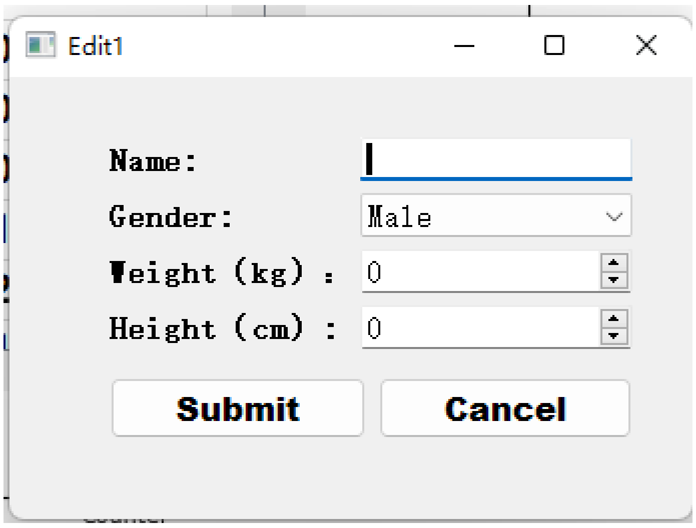
   g. Time 和 Force 会实时显示模拟数据流的输入时间和力的大小

ii. 中上区表格会在数据流输入时自动显示 3 跳的相关参数

iii. 右上区域会动态显示力的曲线

iv. 下方会实时根据输入数据计算出反跳时力随时间变化图，自动标注出相关的数据和阶段，并自动保存在同目录文件夹下。

4. 编辑用户信息界面


Edit1

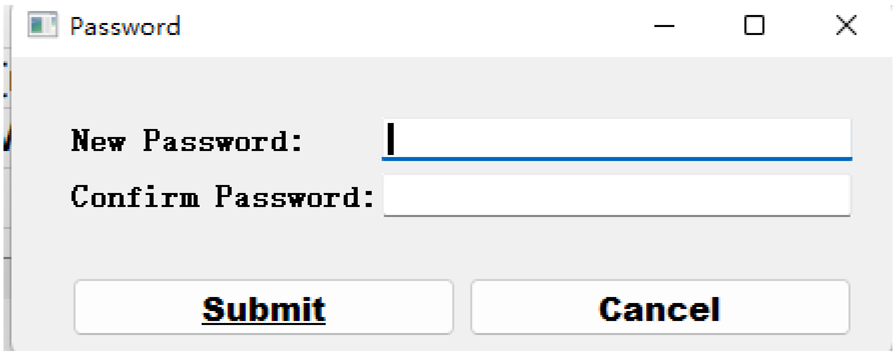
Name:	
Gender:	Male
Weight (kg):	0
Height (cm):	0

在相应栏目中输入相应数值，点击 Submit 即可保存数据，并动态更新到主页面中。

点击 cancel 可以取消编辑

Ps：性别有三个可选项，身高体重有一定的范围限制，保证输入数据的合法性。

5. 修改密码界面


Password

New Password:	
Confirm Password:	

输入密码和确认密码后点击 submit 即可修改密码（如两次密码不一致会有相关提示），点击 Cancel 可以取消修改

【技术文档】

(1) 程序亮点:

1. 用户做完每次反向跳后，程序自动标记出 start/counter movement/braking/push off/takeoff/landing/flight 等
2. 指标数量大于 5 个
3. 登陆注册功能完善，能充分检查输入的合法性，避免因为数据结构问题而出现运行错误。
4. 利用 excel 储存数据，关闭程序后数据不会丢失。
5. 能编辑用户数据和重置密码，功能完善
6. 实时显示力时间参数，并将数据实时可视化。
7. 可以自动截取出每一跳的区间并绘图，自动保存在文件夹中
8. 自动计算出相应参数，并可以导出为 excel 表格

(2) 程序整体结构（模块划分）以及各模块功能描述：

class parameter：用于储存 G 值等常量参数

class io：用于数据的输入和输出（用户数据，实验数据，结果读写）

@classmethod

def read(cls, s)：

读文件，s 传入所需读的是哪个 Excel

@classmethod

def write(cls, s):

写文件，s 传入所需写的是哪个 Excel

class Ui_Login：登陆界面

setupUi(self, Login):面板初始化

retranslateUi(self, Login)：widget 重命名

def check(self):检查用户名密码与输入是否符合，输出相应提示

class Ui_Register：注册界面

CheckRegister(self)：检查注册用户名密码合法性

class MyMplCanvas：画布控件，在主界面中的动画 widget 初始化类

class Ui_MainWindow：主界面

SetFont(self)：插件太多用这个函数批量调整字体

draw(self)：画动态的数据流输入曲线图

diff(self, i)：用于比较间隔时间点的力的差距，i 为第几跳

update_line(self, i)：更新时间窗口，并计算出每一跳的起止时间，i 用于更新时间窗

draw2(self, i)：用于画出每一跳结果的曲线图，i 为第 i 跳

draw_line(self)：在画出的结果图中标注阶段和标记

parameter(self, i)：计算每一跳中的参数，i 为第 i 跳

export_parameter(self, i) 将参数导入到 dataframe 中，i 为第 i 跳

export(self) ：检查数据是否写入 dataframe，将 dataframe 中数据导入 excel。

on_start(self)：开始模拟数据流输入

on_stop(self) ：暂停模拟数据流输入

tableView_init(self, i)：初始化表，i 用于判断初始化哪个表

tableView_add(self, i)：向表中添加数据项，i 用于判断添加到哪个表中

tableView_clear(self)：清空表

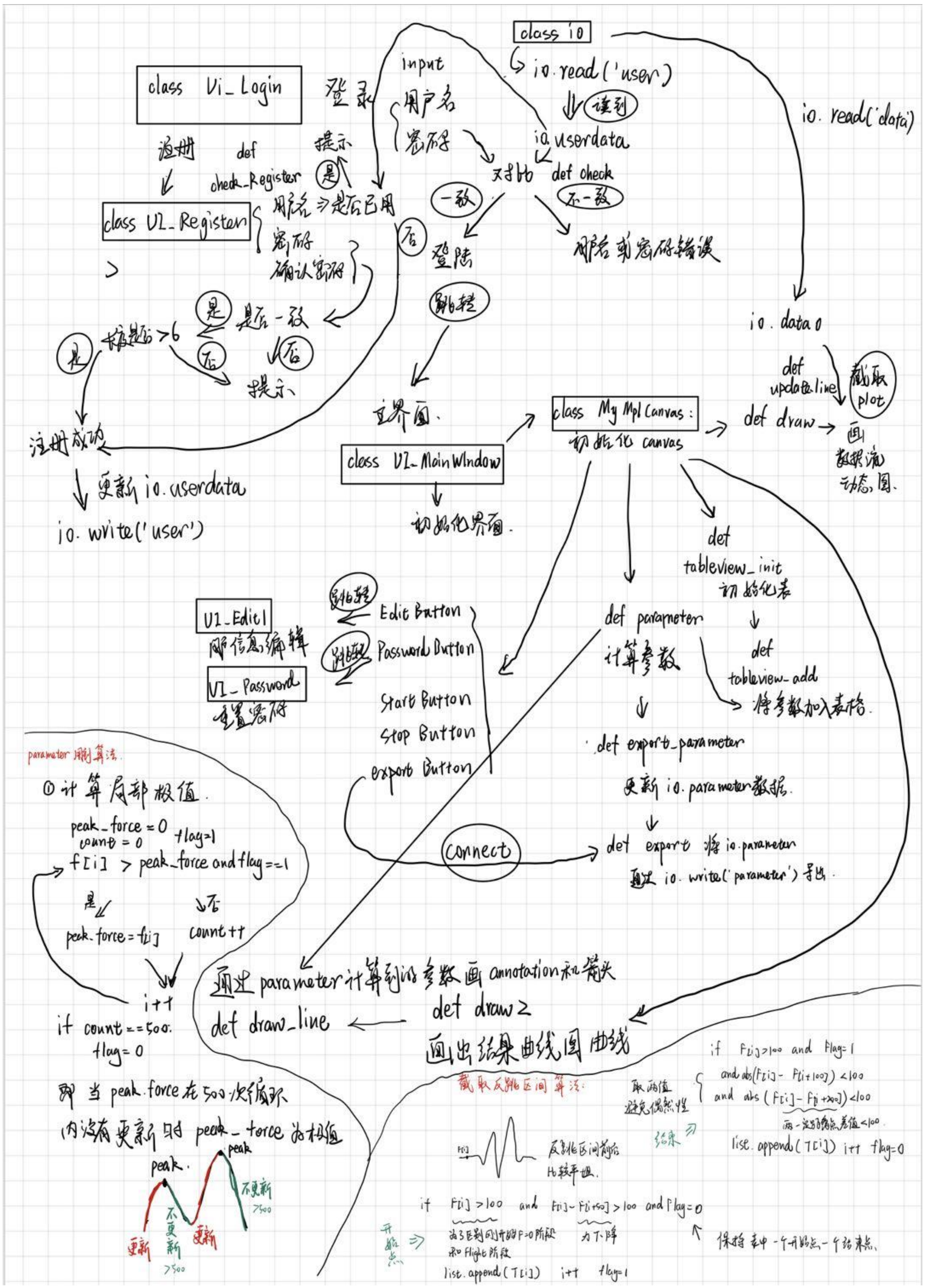
class Ui_Edit1：编辑用户信息界面

Edit(self) ：检查输入合法性，给出相应提示，更新数据到 excel

class Ui_Password：重置密码界面

CheckPassword(self)：检查两次密码是否一致，更新到 excel

⑵　各模块的主要算法（用流程图描述）和数据结构解释：



(3) 选用的测试数据及测试结果

测试数据：详见同一文件夹中的 原始数据.xlsx 数据格式如下：

```
             Time      force
0          0.000 -0.613756
1          0.001 -0.113756
...          ...        ...
29039   29.039 -0.613756
29040   29.040   0.136244
[29041 rows x 2 columns]
```

测试结果：详见同一文件夹中的 parameter.xlsx 数据格式如下：

```
              Body weight[kg]   ...   Start interval A[m/s^2]
0             66.521556   ...                     2.574767
1             66.710841   ...                     1.944684
2             66.327168   ...                     1.244483
[3 rows x 7 columns] (5)
```

力测量分析图片见同文件夹中的 Jump 1.jpg，Jump 2.jpg，Jump 3.jpg

（4）程序的源代码清单：

```
import sys

from PyQt5 import QtWidgets, QtCore, QtGui

import pandas as pd

import numpy as np

from matplotlib.backends.backend_qt4agg import FigureCanvasQTAgg as FigureCanvas

from matplotlib.figure import Figure

from matplotlib.animation import FuncAnimation

import matplotlib.pyplot as plt



class parameter():

    G = 9.8

    Start_interval_Time = 500

    count1 = 1

    update_rate = 10



class io():

    userdata = pd.DataFrame()

    data0 = pd.DataFrame()

    parameter = pd.DataFrame()

    user = ''

    len_user = 0

    userid = 0


    @classmethod

    def read(cls, s):

        if s == 'user':

            io.userdata = pd.read_excel('userdata.xlsx')

            io.user = io.userdata.iloc[:, 0:2]

            io.len_user = io.user.shape[0]

        if s == 'data':
            io.data0 = pd.read_excel('原始数据.xlsx')

    @classmethod
    def write(cls, s):
        if s == 'user':
            io.userdata.to_excel('userdata.xlsx', index=False)
        if s == 'data':
            io.data0.to_excel('userdata.xlsx', index=False)
        if s == 'parameter':
            io.parameter.to_excel('parameter.xlsx', index=False)


class Ui_Login(object):

    def __init__(self):
        self.flag = 0

    def setupUi(self, Login):
        Login.setObjectName("Login")
        Login.resize(600, 400)
        self.centralwidget = QtWidgets.QWidget(Login)
        self.centralwidget.setObjectName("centralwidget")
        self.SignIn = QtWidgets.QPushButton(self.centralwidget)
        self.SignIn.setGeometry(QtCore.QRect(120, 250, 90, 30))
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPointSize(14)
        self.SignIn.setFont(font)
        self.SignIn.setObjectName("SignIn")
        self.Register = QtWidgets.QPushButton(self.centralwidget)
        self.Register.setGeometry(QtCore.QRect(340, 250, 90, 30))
        font = QtGui.QFont()
        font.setFamily("Arial")
font.setPointSize(14)

self.Register.setFont(font)

self.Register.setObjectName("Register")

self.LineEdit = QtWidgets.QLineEdit(self.centralwidget)

self.LineEdit.setGeometry(QtCore.QRect(200, 90, 270, 30))

font = QtGui.QFont()

font.setFamily("Arial")

font.setPointSize(14)

self.LineEdit.setFont(font)

self.LineEdit.setObjectName("LineEdit")

self.LineEdit_2 = QtWidgets.QLineEdit(self.centralwidget)

self.LineEdit_2.setGeometry(QtCore.QRect(200, 160, 270, 30))

font = QtGui.QFont()

font.setFamily("Arial")

font.setPointSize(14)

self.LineEdit_2.setFont(font)

self.LineEdit_2.setObjectName("LineEdit_2")

self.label = QtWidgets.QLabel(self.centralwidget)

self.label.setGeometry(QtCore.QRect(80, 90, 90, 30))

font = QtGui.QFont()

font.setFamily("Arial")

font.setPointSize(14)

self.label.setFont(font)

self.label.setObjectName("label")

self.label_2 = QtWidgets.QLabel(self.centralwidget)

self.label_2.setGeometry(QtCore.QRect(80, 160, 90, 30))

font = QtGui.QFont()

font.setFamily("Arial")

font.setPointSize(14)

self.label_2.setFont(font)

self.label_2.setObjectName("label_2")

self.label_3 = QtWidgets.QLabel(self.centralwidget)

self.label_3.setGeometry(QtCore.QRect(50, 210, 500, 21))
```

```
        palette = QtGui.QPalette()
        brush = QtGui.QBrush(QtGui.QColor(255, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
        self.label_3.setPalette(palette)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPointSize(14)
        self.label_3.setFont(font)
        self.label_3.setText("")
        self.label_3.setObjectName("label_3")
        Login.setCentralWidget(self.centralwidget)
        self.retranslateUi(Login)
        QtCore.QMetaObject.connectSlotsByName(Login)

        self.SignIn.clicked.connect(self.check)

    def retranslateUi(self, Login):
        _translate = QtCore.QCoreApplication.translate
        Login.setWindowTitle(_translate("Login", "MainWindow"))
        self.SignIn.setText(_translate("Login", "Sign In"))
        self.Register.setText(_translate("Login", "Register"))
        self.label.setText(_translate("Login", "Username:"))
        self.label_2.setText(_translate("Login", "Password:"))

    def check(self):
        username = self.LineEdit.text()
        password = self.LineEdit_2.text()

        io.read('user')
        flag1 = 1
        for i in range(io.len_user):
            if io.user.iloc[i].user == username and io.user.iloc[i].password == password:
                self.flag = 1

                flag1 = 0

                self.label_3.setGeometry(QtCore.QRect(230, 210, 500, 21))

                io.userid = i

                Name = io.userdata.iloc[io.userid, 2]

                print(Name)

                self.label_3.setText(f'Welcome.{Name}')


        if flag1 == 1:

            self.label_3.setText('Your username or password is incorrect. Please try again.')




class Ui_Register(object):

    def setupUi(self, Register):

        Register.setObjectName("Register")

        Register.resize(600, 400)

        self.centralwidget = QtWidgets.QWidget(Register)

        self.centralwidget.setObjectName("centralwidget")

        self.Register = QtWidgets.QPushButton(self.centralwidget)

        self.Register.setGeometry(QtCore.QRect(120, 320, 90, 30))

        font = QtGui.QFont()

        font.setFamily("Arial")

        font.setPointSize(14)

        self.Register.setFont(font)

        self.Register.setObjectName("Register")

        self.Quit = QtWidgets.QPushButton(self.centralwidget)

        self.Quit.setGeometry(QtCore.QRect(340, 320, 90, 30))

        font = QtGui.QFont()

        font.setFamily("Arial")

        font.setPointSize(14)

        self.Quit.setFont(font)

        self.Quit.setObjectName("Quit")

        self.LineEdit = QtWidgets.QLineEdit(self.centralwidget)
self.LineEdit.setGeometry(QtCore.QRect(300, 90, 270, 30))
font = QtGui.QFont()
font.setFamily("Arial")
font.setPointSize(14)
self.LineEdit.setFont(font)
self.LineEdit.setObjectName("LineEdit")
self.LineEdit_2 = QtWidgets.QLineEdit(self.centralwidget)
self.LineEdit_2.setGeometry(QtCore.QRect(300, 160, 270, 30))
font = QtGui.QFont()
font.setFamily("Arial")
font.setPointSize(14)
self.LineEdit_2.setFont(font)
self.LineEdit_2.setObjectName("LineEdit_2")
self.LineEdit_3 = QtWidgets.QLineEdit(self.centralwidget)
self.LineEdit_3.setGeometry(QtCore.QRect(300, 230, 270, 30))
font = QtGui.QFont()
font.setFamily("Arial")
font.setPointSize(14)
self.LineEdit_3.setFont(font)
self.LineEdit_3.setObjectName("LineEdit_3")
self.label = QtWidgets.QLabel(self.centralwidget)
self.label.setGeometry(QtCore.QRect(80, 90, 200, 30))
font = QtGui.QFont()
font.setFamily("Arial")
font.setPointSize(14)
self.label.setFont(font)
self.label.setObjectName("label")
self.label_2 = QtWidgets.QLabel(self.centralwidget)
self.label_2.setGeometry(QtCore.QRect(80, 160, 200, 30))
font = QtGui.QFont()
font.setFamily("Arial")
font.setPointSize(14)
self.label_2.setFont(font)
```

```
        self.label_2.setObjectName("label_2")
        self.label_3 = QtWidgets.QLabel(self.centralwidget)
        self.label_3.setGeometry(QtCore.QRect(80, 230, 200, 30))
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPointSize(14)
        self.label_3.setFont(font)
        self.label_3.setObjectName("label_3")
        self.label_4 = QtWidgets.QLabel(self.centralwidget)
        self.label_4.setGeometry(QtCore.QRect(100, 280, 500, 21))
        palette = QtGui.QPalette()
        brush = QtGui.QBrush(QtGui.QColor(255, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 0, 0, 128))
        brush.setStyle(QtCore.Qt.SolidPattern)
        self.label_4.setPalette(palette)
        font = QtGui.QFont()
        font.setFamily("Arial")
        font.setPointSize(14)
        self.label_4.setFont(font)
        self.label_4.setText(" ")
        self.label_4.setObjectName("label_4")
        Register.setCentralWidget(self.centralwidget)
        self.retranslateUi(Register)
        QtCore.QMetaObject.connectSlotsByName(Register)

        self.Register.clicked.connect(self.CheckRegister)

    def retranslateUi(self, Login):
        _translate = QtCore.QCoreApplication.translate
        Login.setWindowTitle(_translate("Login", "MainWindow"))
        self.Register.setText(_translate("Login", "Register"))
        self.Quit.setText(_translate("Login", "Cancel"))
        self.label.setText(_translate("Login", "New Username:"))
        self.label_2.setText(_translate("Login", "New Password:"))
        self.label_3.setText(_translate("Login", "Retry new Password:"))


    def CheckRegister(self):
        username = self.LineEdit.text()
        password = self.LineEdit_2.text()
        confirm_password = self.LineEdit_3.text()
        io.read('user')
        if password != confirm_password:
            self.label_4.setText('Password and confirm password do not match.')
        else:
            flag2 = 1
            for i in range(io.len_user):
                if io.userdata.iloc[i, 0] == username:
                    self.label_4.setGeometry(QtCore.QRect(180, 280, 300, 21))
                    self.label_4.setText('This username is occupied')
                    flag2 = 0
                    break
            if flag2 == 1 and len(password) < 6:
                self.label_4.setGeometry(QtCore.QRect(130, 280, 500, 21))
                self.label_4.setText('Password must be at least 6 characters. ')
                flag2 = 0
            if flag2 == 1:
                self.label_4.setGeometry(QtCore.QRect(200, 280, 300, 21))
                self.label_4.setText('Registration successful.')
                io.userdata.loc[io.len_user] = [username, password, 'Unknown', 'Unknown', 'Unknown',
'Unknown',
                                                'Unknown']
                print(io.userdata)
                io.write('user')


class MyMplCanvas(FigureCanvas):
    """创建了一个 widget"""

    def __init__(self, parent=None, width=5, height=4, dpi=100):
        fig = Figure(figsize=(width, height), dpi=dpi)
        self.axes = fig.add_subplot(111)

        self.compute_initial_figure()

        FigureCanvas.__init__(self, fig)
        self.setParent(parent)

    def compute_initial_figure(self):
        pass


class Ui_MainWindow(QtWidgets.QWidget):
    flag = 0

    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1800, 1000)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")

        self.verticalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.verticalLayoutWidget.setGeometry(QtCore.QRect(20, 20, 350, 500))
        self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")

        self.StartButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.StartButton.setFont(self.SetFont())
        self.StartButton.setObjectName("StartButton")
        self.verticalLayout.addWidget(self.StartButton)
        self.StartButton.clicked.connect(self.on_start)

        self.StopButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.StopButton.setFont(self.SetFont())
        self.StopButton.setObjectName("StopButton")
        self.verticalLayout.addWidget(self.StopButton)
        self.StopButton.clicked.connect(self.on_stop)

        self.tableView_init(0)

        self.EditButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.EditButton.setFont(self.SetFont())
        self.EditButton.setObjectName("EditButton")
        self.verticalLayout.addWidget(self.EditButton)

        self.PasswordButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.PasswordButton.setFont(self.SetFont())
        self.PasswordButton.setObjectName("PasswordButton")
        self.verticalLayout.addWidget(self.PasswordButton)

        self.ExportParameterButton = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.ExportParameterButton.setFont(self.SetFont())
        self.ExportParameterButton.setObjectName("ExportParameterButton")
        self.verticalLayout.addWidget(self.ExportParameterButton)
        self.ExportParameterButton.clicked.connect(self.export)

        self.horizontalLayoutWidget_3 = QtWidgets.QWidget(self.centralwidget)
        self.horizontalLayoutWidget_3.setGeometry(QtCore.QRect(20, 540, 1760, 440))

        self.horizontalLayoutWidget_3.setObjectName("horizontalLayoutWidget_3")

        self.horizontalLayout_3 = QtWidgets.QHBoxLayout(self.horizontalLayoutWidget_3)

        self.horizontalLayout_3.setContentsMargins(0, 0, 0, 0)

        self.horizontalLayout_3.setObjectName("horizontalLayout_3")


        self.label1 = QtWidgets.QLabel("label1", self)

        self.label1.setText('Time: 0.000         Force: 0.000')

        self.label1.setFont(self.SetFont())


        self.verticalLayout.addWidget(self.label1)


        self.verticalLayoutWidget_2 = QtWidgets.QWidget(self.centralwidget)

        self.verticalLayoutWidget_2.setGeometry(QtCore.QRect(380, 20, 480, 500))

        self.verticalLayoutWidget_2.setObjectName("verticalLayoutWidget_2")

        self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.verticalLayoutWidget_2)

        self.verticalLayout_2.setContentsMargins(0, 0, 0, 0)

        self.verticalLayout_2.setObjectName("verticalLayout_2")


        self.tableView_init(1)


        self.verticalLayoutWidget_3 = QtWidgets.QWidget(self.centralwidget)

        self.verticalLayoutWidget_3.setGeometry(QtCore.QRect(880, 20, 890, 500))

        self.verticalLayoutWidget_3.setObjectName("verticalLayoutWidget_3")

        self.verticalLayout_3 = QtWidgets.QVBoxLayout(self.verticalLayoutWidget_3)

        self.verticalLayout_3.setContentsMargins(0, 0, 0, 0)

        self.verticalLayout_3.setObjectName("verticalLayout_3")


        self.draw()


        MainWindow.setCentralWidget(self.centralwidget)


        self.retranslateUi(MainWindow)
```

```
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def SetFont(self):
        font = QtGui.QFont()
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        return font

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.StartButton.setText(_translate("MainWindow", "Input Start"))
        self.StopButton.setText(_translate("MainWindow", "Input Pause"))
        self.EditButton.setText(_translate("MainWindow", "Edit Profile"))
        self.PasswordButton.setText(_translate("MainWindow", "Change Password"))
        self.ExportParameterButton.setText(_translate("MainWindow", "Export Parameter"))

    def draw(self):
        self.canvas = MyMplCanvas(self, width=8, height=6, dpi=100)
        self.verticalLayout_3.addWidget(self.canvas)
        self.p = 0

        io.read('data')
        self.x0 = io.data0.Time
        self.x = np.linspace(0, 100, 100)
        self.y0 = io.data0.force
        self.y = np.linspace(-1, 4000, 100)
        self.line, = self.canvas.axes.plot(self.x, self.y, animated=True, lw=1)

        self.flag = 0
        self.listP = []
        self.body_weight = 0
        self.i = 0

    def diff(self, i):
        return abs(self.y0[self.p] - self.y0[self.p + i]) < 100

    def update_line(self, i):
        if (120 + self.p) <= io.data0.shape[0]:
            self.p += parameter.update_rate
        y = self.y0[self.p:100 + self.p]
        t = self.x0[self.p]
        f = self.y0[self.p]

        if self.flag == 0 and self.y0[self.p] > 100 and (self.y0[self.p] - self.y0[self.p + 50]) > 100:
            self.listP.append(self.p)
            self.i += 1
            self.flag = 1

        if self.flag == 1 and self.y0[self.p] > 100 and self.diff(200) and self.diff(100) and self.diff(50):
            self.listP.append(self.p)
            self.draw2(self.i)
            self.flag = 0

        self.line.set_ydata(y)
        self.label1.setText('Time: %5.3f        Force: %5.3f ' % (t, f))
        return [self.line]

    def draw2(self, i):
        self.x = self.x0[self.listP[2 * i - 2] - 200:self.listP[2 * i - 1] + 200]
        self.y = self.y0[self.listP[2 * i - 2] - 200:self.listP[2 * i - 1] + 200]
        self.canvas2 = FigureCanvas(plt.Figure(figsize=(8, 6)))
        self.ax = self.canvas2.figure.subplots()
        self.horizontalLayout_3.addWidget(self.canvas2)
        self.ax.plot(self.x, self.y)
        self.parameter(i)
        self.draw_line()
        self.tableView_add(1)
        self.export_parameter(i)

        self.ax.set_title(f'Jump {i}')
        self.ax.set_xlabel('Time[s]')
        self.ax.set_ylabel('Ground reaction force[N]')
        self.canvas2.figure.savefig(f'Jump {i}.jpg')

    def draw_line(self):
        self.take_off_time = self.flight.iloc[0, 0]
        self.landing_time = self.flight.iloc[-1, 0]
        self.start = self.period.iloc[0, 0]

        self.ax.plot([self.start - 0.02] * 2, [self.period.iloc[0, 1], 2700], '--', color='gray', lw=1)
        self.ax.annotate('Start', xy=(self.start - 0.02, self.period.iloc[0, 1]), fontsize=8,
                         xytext=(self.start - 0.15, self.period.iloc[0, 1] - 500),
                         arrowprops=dict(facecolor='gray', arrowstyle='->'))

        self.ax.plot([self.take_off_time] * 2, [0, 2700], '--', color='gray', lw=1)
        self.ax.annotate('Take Off', xy=(self.take_off_time, 0), fontsize=8,
                         xytext=(self.take_off_time - 0.40, 100),
                         arrowprops=dict(facecolor='gray', arrowstyle='->'))

        self.ax.plot([self.landing_time] * 2, [0, 2700], '--', color='gray', lw=1)
        self.ax.annotate('Landing', xy=(self.landing_time, 0), fontsize=8,
                         xytext=(self.landing_time + 0.15, 100),
                         arrowprops=dict(facecolor='gray', arrowstyle='->'))

        self.ax.plot([self.low_index / 1000] * 2, [self.low_force, 2300], '--', color='gray', lw=1)
        self.ax.plot([self.peak_index / 1000] * 2, [self.peak_force, 2700], '--', color='gray', lw=1)
        self.ax.annotate('', xy=(self.start - 0.02, 2700),
                         xytext=(self.peak_index / 1000, 2700),
                         arrowprops=dict(facecolor='gray', arrowstyle='<->'))
        self.ax.annotate('', xy=(self.peak_index / 1000, 2700),
                         xytext=(self.take_off_time, 2700),
                         arrowprops=dict(facecolor='gray', arrowstyle='<->'))
        self.ax.annotate('', xy=(self.take_off_time, 2700),
                         xytext=(self.landing_time, 2700),
                         arrowprops=dict(facecolor='gray', arrowstyle='<->'))
        self.ax.annotate('', xy=(self.low_index / 1000, 2300),
                         xytext=(self.peak_index / 1000, 2300),
                         arrowprops=dict(facecolor='gray', arrowstyle='<->'))
        self.ax.annotate('Braking', xy=(self.peak_index / 1000 - 0.2, 2400), fontsize=8)
        self.ax.annotate('Counter₩nMovement', xy=(self.peak_index / 1000 - 0.33, 2800), fontsize=8)
        self.ax.annotate('Push Off', xy=(self.take_off_time - 0.26, 2800), fontsize=8)
        self.ax.annotate('Flight', xy=(self.landing_time - 0.27, 2800), fontsize=8)

    def parameter(self, i):
        self.flag2 = 0
        self.flag3 = 0
        self.t1 = 0
        self.t2 = 0
        self.peak_force = 0
        self.peak_index = 0
        self.low_force = 1000
        self.low_index = 0
        self.last_peak_time = 0
        self.count1 = 0
        self.count2 = 0
        self.body_weight = np.average(io.data0.iloc[self.listP[2 * i - 2] - 200:self.listP[2 * i - 2] - 100, 1])
        self.period = io.data0[self.listP[2 * i - 2]:self.listP[2 * i - 1]]
        for i in range(self.listP[2 * i - 2], self.listP[2 * i - 1]):
            f = io.data0.iloc[i, 1]
            t = io.data0.iloc[i, 0]
            if f > self.peak_force and self.flag2 == 0:
                self.peak_force = f
                self.peak_index = i
            else:
                self.count1 += 1
            if self.count1 == 500:
                self.flag2 = 1

            if f < self.low_force and self.flag3 == 0:
                self.low_force = f
                self.low_index = i
            else:
                self.count2 += 1
            if self.count2 == 100:
                self.flag3 = 1
        print(self.low_force)

        self.body_mass = self.body_weight / parameter.G
        self.flight = self.period[self.period.force < 4]
        self.flight_T = self.flight.shape[0] / 1000
        self.jump_T = self.flight.iloc[0, 0] - self.period.iloc[0, 0]
        self.jump_height_from_flight_T = parameter.G * (self.flight_T / 2) ** 2 / 2
        self.start_interval = io.data0[self.peak_index:self.peak_index + parameter.Start_interval_Time]
        self.Start_interval_relative_F = np.average(self.start_interval.iloc[:, 1] / self.body_weight)
        self.relative_max_F = self.peak_force / self.body_weight
        self.start_interval_A = np.average(self.start_interval.iloc[:, 1] / self.body_mass) - parameter.G

    def export_parameter(self, i):
        if i == 1:
            io.parameter = pd.DataFrame(columns=self.parameter_list)
        io.parameter.loc[i - 1] = self.value_list

    def export(self):
        if io.parameter.shape[1] == 7:
            io.write('parameter')

    def on_start(self):
        if Ui_MainWindow.flag == 0:
            self.ani = FuncAnimation(self.canvas.figure, self.update_line, frames=10, blit=True, interval=0)
            Ui_MainWindow.flag = 1

    def on_stop(self):
        if Ui_MainWindow.flag == 1:
            self.ani._stop()
            Ui_MainWindow.flag = 0

    def tableView_init(self, i):
        if i == 0:
            # 创建一个 0 行 2 列 的标准模型
            self.model = QtGui.QStandardItemModel(0, 2)

            self.tableview = QtWidgets.QTableView(self.verticalLayoutWidget)
            self.tableview.setFont(self.SetFont())
            self.tableview.setObjectName("tableView")
            self.verticalLayout.addWidget(self.tableview)

            # 创建 tableView 组件
            # 将 tableView 添加到垂直盒布局里

            # tableView 组件 设置模型
            self.tableview.setModel(self.model)

            self.tableview.horizontalHeader().setSectionResizeMode(QtWidgets.QHeaderView.Stretch)  # 所有列
自动拉伸，充满界面
```

```
            # self.tableview.setSelectionMode(QtWidgets.QAbstractItemView.SingleSelection)  # 设置只能选中
整行
            self.tableview.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)  # 不可编辑
            self.tableview.setSelectionBehavior(QtWidgets.QAbstractItemView.SelectRows)
            # 设置只能选中一行

            # self.tableView.setSelectionBehavior(QAbstractItemView.SelectRows)  # 设置只能选中整行
            # self.tableView.setSelectionMode(QAbstractItemView.ExtendedSelection)  # 设置只能选中多行
        if i == 1:
            self.model2 = QtGui.QStandardItemModel(0, 2)
            self.model2.setHorizontalHeaderLabels(['parameter', 'value'])

            self.tableview2 = QtWidgets.QTableView(self.verticalLayoutWidget_2)
            self.tableview2.setFont(self.SetFont())
            self.tableview2.setObjectName("tableview2")
            self.verticalLayout_2.addWidget(self.tableview2)
            self.tableview2.setModel(self.model2)
            self.tableview2.horizontalHeader().setSectionResizeMode(QtWidgets.QHeaderView.Stretch)  # 所有
列自动拉伸，充满界面
            self.tableview.setSelectionMode(QtWidgets.QAbstractItemView.SingleSelection)  # 设置只能选中整
行
            self.tableview2.setEditTriggers(QtWidgets.QAbstractItemView.NoEditTriggers)  # 不可编辑
            self.tableview2.setSelectionBehavior(QtWidgets.QAbstractItemView.SelectRows)

            # self.tableView.setSelectionBehavior(QAbstractItemView.SelectRows)  # 设置只能选中整行
            # self.tableView.setSelectionMode(QAbstractItemView.ExtendedSelection)  # 设置只能选中多行


    def tableView_add(self, i):
        if i == 0:
            io.read('user')
            for i in range(2, 7):
                self.model.appendRow([
                    QtGui.QStandardItem(io.userdata.columns[i]),
                    QtGui.QStandardItem(f'{io.userdata.iloc[io.userid, i]}')
                ])
        if i == 1:

            self.parameter_list = ['Body weight[kg]', 'Flight T[s]',
                                   'Jump T[s]', 'Jump Height from Flight T[m]',
                                   'Start interval relative F[%BW] ', 'Relative maximal F[%BW] ',
                                   'Start interval A[m/s^2] ']
            self.value_list = [self.body_mass,
                               self.flight_T,
                               self.jump_T,
                               self.jump_height_from_flight_T,
                               self.Start_interval_relative_F,
                               self.relative_max_F,
                               self.start_interval_A]
            self.model2.appendRow([
                QtGui.QStandardItem(f'————————jump {parameter.count1}———————'),
                QtGui.QStandardItem('———————')
            ])
            parameter.count1 += 1
            for i in range(0, 7):
                self.model2.appendRow([
                    QtGui.QStandardItem(self.parameter_list[i]),
                    QtGui.QStandardItem(f'{round(self.value_list[i], 3)}')
                ])
            self.tableview2.horizontalHeader().setSectionResizeMode(QtWidgets.QHeaderView.ResizeToContents)

    def tableView_clear(self):

        self.model.clear()


class Ui_Edit1(object):
    def setupUi(self, Edit1):
        Edit1.setObjectName("Edit1")
```

```
Edit1.resize(341, 221)

self.centralwidget = QtWidgets.QWidget(Edit1)

self.centralwidget.setObjectName("centralwidget")

self.formLayoutWidget = QtWidgets.QWidget(self.centralwidget)

self.formLayoutWidget.setGeometry(QtCore.QRect(50, 30, 261, 111))

self.formLayoutWidget.setObjectName("formLayoutWidget")

self.formLayout = QtWidgets.QFormLayout(self.formLayoutWidget)

self.formLayout.setContentsMargins(0, 0, 0, 0)

self.formLayout.setHorizontalSpacing(2)

self.formLayout.setObjectName("formLayout")


self.label = QtWidgets.QLabel(self.formLayoutWidget)

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label.setFont(font)

self.label.setObjectName("label")

self.formLayout.setWidget(0, QtWidgets.QFormLayout.LabelRole, self.label)


self.label_2 = QtWidgets.QLabel(self.formLayoutWidget)

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label_2.setFont(font)

self.label_2.setObjectName("label_2")

self.formLayout.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.label_2)


self.label_3 = QtWidgets.QLabel(self.formLayoutWidget)

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)
font.setWeight(75)
self.label_3.setFont(font)
self.label_3.setObjectName("label_3")
self.formLayout.setWidget(2, QtWidgets.QFormLayout.LabelRole, self.label_3)

self.label_4 = QtWidgets.QLabel(self.formLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.label_4.setFont(font)
self.label_4.setObjectName("label_4")
self.formLayout.setWidget(3, QtWidgets.QFormLayout.LabelRole, self.label_4)

self.LineEdit = QtWidgets.QLineEdit(self.formLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
self.LineEdit.setFont(font)
self.LineEdit.setObjectName("LineEdit")
self.formLayout.setWidget(0, QtWidgets.QFormLayout.FieldRole, self.LineEdit)

self.comboBox = QtWidgets.QComboBox(self.formLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
self.comboBox.setFont(font)
self.comboBox.setObjectName("comboBox")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.comboBox.addItem("")
self.formLayout.setWidget(1, QtWidgets.QFormLayout.FieldRole, self.comboBox)

self.spinBox = QtWidgets.QSpinBox(self.formLayoutWidget)
font = QtGui.QFont()
```

```
font.setPointSize(12)
self.spinBox.setFont(font)
self.spinBox.setMaximum(999)
self.spinBox.setObjectName("spinBox")
self.formLayout.setWidget(2, QtWidgets.QFormLayout.FieldRole, self.spinBox)

self.spinBox_2 = QtWidgets.QSpinBox(self.formLayoutWidget)
font = QtGui.QFont()
font.setPointSize(12)
self.spinBox_2.setFont(font)
self.spinBox_2.setMaximum(300)
self.spinBox_2.setObjectName("spinBox_2")
self.formLayout.setWidget(3, QtWidgets.QFormLayout.FieldRole, self.spinBox_2)

self.horizontalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
self.horizontalLayoutWidget.setGeometry(QtCore.QRect(50, 150, 261, 31))
self.horizontalLayoutWidget.setObjectName("horizontalLayoutWidget")
self.horizontalLayout = QtWidgets.QHBoxLayout(self.horizontalLayoutWidget)
self.horizontalLayout.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout.setObjectName("horizontalLayout")

self.pushButton = QtWidgets.QPushButton(self.horizontalLayoutWidget)
font = QtGui.QFont()
font.setFamily("Arial Black")
font.setPointSize(12)
font.setBold(True)
font.setWeight(75)
self.pushButton.setFont(font)
self.pushButton.setObjectName("pushButton")
self.horizontalLayout.addWidget(self.pushButton)
self.pushButton.clicked.connect(self.Edit)

self.pushButton_2 = QtWidgets.QPushButton(self.horizontalLayoutWidget)
```

```
        font = QtGui.QFont()
        font.setFamily("Arial Black")
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setObjectName("pushButton_2")
        self.horizontalLayout.addWidget(self.pushButton_2)

        Edit1.setCentralWidget(self.centralwidget)

        self.retranslateUi(Edit1)
        QtCore.QMetaObject.connectSlotsByName(Edit1)

    def retranslateUi(self, Edit1):
        _translate = QtCore.QCoreApplication.translate
        Edit1.setWindowTitle(_translate("Edit1", "Edit1"))
        self.label.setText(_translate("Edit1", "Name:"))
        self.label_2.setText(_translate("Edit1", "Gender:"))
        self.label_3.setText(_translate("Edit1", "Weight ( kg )  : "))
        self.label_4.setText(_translate("Edit1", "Height ( cm ) :"))
        self.comboBox.setItemText(0, _translate("Edit1", "Male"))
        self.comboBox.setItemText(1, _translate("Edit1", "Female"))
        self.comboBox.setItemText(2, _translate("Edit1", "Other Gender"))
        self.pushButton.setText(_translate("Edit1", "Submit"))
        self.pushButton_2.setText(_translate("Edit1", "Cancel"))

    def Edit(self):
        name = self.LineEdit.text()
        gender = self.comboBox.currentText()
        weight = self.spinBox.text()
        height = self.spinBox_2.text()
        BMI = round(eval(weight) / (eval(height) ** 2 / 10000), 2)
        io.userdata.iloc[io.userid, 2:7] = [name, gender, weight, height, BMI]
        print(io.userdata)
        io.write('user')


class Ui_Password(object):
    def setupUi(self, Password):
        Password.setObjectName("Password")
        Password.resize(440, 150)
        self.centralwidget = QtWidgets.QWidget(Password)
        self.centralwidget.setObjectName("centralwidget")
        self.formLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.formLayoutWidget.setGeometry(QtCore.QRect(30, 30, 390, 160))
        self.formLayoutWidget.setObjectName("formLayoutWidget")

        self.formLayout = QtWidgets.QFormLayout(self.formLayoutWidget)
        self.formLayout.setContentsMargins(0, 0, 0, 0)
        self.formLayout.setHorizontalSpacing(2)
        self.formLayout.setObjectName("formLayout")

        self.label = QtWidgets.QLabel(self.formLayoutWidget)
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.formLayout.setWidget(0, QtWidgets.QFormLayout.LabelRole, self.label)

        self.LineEdit = QtWidgets.QLineEdit(self.formLayoutWidget)
        font = QtGui.QFont()
        font.setPointSize(12)
        self.LineEdit.setFont(font)
self.LineEdit.setObjectName("LineEdit")

self.formLayout.setWidget(0, QtWidgets.QFormLayout.FieldRole, self.LineEdit)


self.label_2 = QtWidgets.QLabel(self.formLayoutWidget)

font = QtGui.QFont()

font.setPointSize(12)

font.setBold(True)

font.setWeight(75)

self.label_2.setFont(font)

self.label_2.setObjectName("label_2")

self.formLayout.setWidget(1, QtWidgets.QFormLayout.LabelRole, self.label_2)


self.LineEdit_2 = QtWidgets.QLineEdit(self.formLayoutWidget)

font = QtGui.QFont()

font.setPointSize(12)

self.LineEdit_2.setFont(font)

self.LineEdit_2.setObjectName("LineEdit2")

self.formLayout.setWidget(1, QtWidgets.QFormLayout.FieldRole, self.LineEdit_2)


self.label_3 = QtWidgets.QLabel(self.centralwidget)

self.label_3.setGeometry(QtCore.QRect(30, 80, 390, 20))

palette = QtGui.QPalette()

brush = QtGui.QBrush(QtGui.QColor(255, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 0, 0))

brush.setStyle(QtCore.Qt.SolidPattern)

palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)

brush = QtGui.QBrush(QtGui.QColor(255, 0, 0, 128))

brush.setStyle(QtCore.Qt.SolidPattern)

self.label_3.setPalette(palette)

font = QtGui.QFont()

font.setPointSize(12)
font.setBold(True)

font.setWeight(75)

self.label_3.setFont(font)

self.label_3.setObjectName("label_3")

self.label_3.setText(" ")


self.horizontalLayoutWidget = QtWidgets.QWidget(self.centralwidget)

self.horizontalLayoutWidget.setGeometry(QtCore.QRect(30, 110, 390, 30))

self.horizontalLayoutWidget.setObjectName("horizontalLayoutWidget")


self.horizontalLayout_2 = QtWidgets.QHBoxLayout(self.horizontalLayoutWidget)

self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)

self.horizontalLayout_2.setObjectName("horizontalLayout_2")


self.pushButton = QtWidgets.QPushButton(self.horizontalLayoutWidget)

font = QtGui.QFont()

font.setFamily("Arial Black")

font.setPointSize(12)

font.setBold(True)

font.setUnderline(True)

font.setWeight(75)

self.pushButton.setFont(font)

self.pushButton.setObjectName("pushButton")

self.horizontalLayout_2.addWidget(self.pushButton)


self.pushButton.clicked.connect(self.CheckPassword)


self.pushButton_2 = QtWidgets.QPushButton(self.horizontalLayoutWidget)

font = QtGui.QFont()

font.setFamily("Arial Black")

font.setPointSize(12)

self.pushButton_2.setFont(font)

self.pushButton_2.setObjectName("pushButton_2")
```

```
        self.horizontalLayout_2.addWidget(self.pushButton_2)
        Password.setCentralWidget(self.centralwidget)

        self.retranslateUi(Password)
        QtCore.QMetaObject.connectSlotsByName(Password)

    def retranslateUi(self, Password):
        _translate = QtCore.QCoreApplication.translate
        Password.setWindowTitle(_translate("Password", "Password"))
        self.label.setText(_translate("Password", "New Password: "))
        self.label_2.setText(_translate("Password", "Confirm Password:"))
        self.pushButton.setText(_translate("Password", "Submit"))
        self.pushButton_2.setText(_translate("Password", "Cancel"))

    def CheckPassword(self):
        password = self.LineEdit.text()
        confirm_password = self.LineEdit_2.text()
        if password != confirm_password:
            self.label_3.setText('Password and confirm password do not match.')

        elif len(password) < 6:
            self.label_3.setGeometry(QtCore.QRect(40, 80, 390, 20))
            self.label_3.setText('Password must be at least 6 characters. ')
        else:
            self.label_3.setGeometry(QtCore.QRect(120, 80, 390, 20))
            self.label_3.setText('Reset password success.')
            io.userdata.iloc[io.userid, 1] = password
            io.write('user')


if __name__ == "__main__":
    import sys
```

```
app = QtWidgets.QApplication(sys.argv)

Login = QtWidgets.QMainWindow()
ui1 = Ui_Login()
ui1.setupUi(Login)

Register = QtWidgets.QMainWindow()


def check():
    i = ui1.flag
    if i:
        Main.show()
        ui3.tableView_clear()
        ui3.tableView_add(0)


def update():
    ui3.tableView_clear()
    ui3.tableView_add(0)


ui2 = Ui_Register()
ui2.setupUi(Register)

Main = QtWidgets.QMainWindow()
ui3 = Ui_MainWindow()
ui3.setupUi(Main)

Edit1 = QtWidgets.QMainWindow()
ui4 = Ui_Edit1()
ui4.setupUi(Edit1)
```

```
Password = QtWidgets.QMainWindow()

ui5 = Ui_Password()

ui5.setupUi(Password)

Login.show()


ui1.Register.clicked.connect(Register.show)

ui2.Quit.clicked.connect(Register.hide)

ui1.SignIn.clicked.connect(check)

ui3.EditButton.clicked.connect(Edit1.show)

ui3.PasswordButton.clicked.connect(Password.show)

ui4.pushButton_2.clicked.connect(Edit1.hide)

ui4.pushButton.clicked.connect(update)

ui5.pushButton_2.clicked.connect(Password.hide)


sys.exit(app.exec_())
```